Minimizing Hamming Weight Based on 1's Complement of Binary Numbers Over GF(2^m)

Xu Huang, Pritam Gajkumar Shah, Dharmendra Sharma
Faculty of Information Sciences and Engineering, University of Canberra, ACT 2601, Australia
Xu.Huang@canberra.edu.au, pritamgshah@yahoo.com, Dharmendra.Sharma@canberra.edu.au

***Abstract*—Elliptic curve cryptosystems have been the focus of much attention as the benefits of elliptic curve cryptography (ECC) become many such as a small software footprint, low hardware implementation costs, linear scalability, low bandwidth requirements, and high performance, which have been drawn great attentions in particular wireless sensor networks. Many papers have investigated various algorithms for fast calculations due to the wireless sensor networks are always limited power energy, constrict computing capacity, and other tighten resources such as storage capacity limited, etc. In this paper a novel algorithm is first presented, with which the hamming weight will be minimized therefore the calculation cost will be dropped and the cryptographic algorithm has gained the natures of ECC. This makes ECC more suitable for use in constrained environment such as mobile sensor information applications, where computing resources and power availability are limited. The final results show that, in comparison with popular algorithms, such as NAF, MOF and complementary algorithms, the proposed algorithm significantly improved (average about 12.5% decreasing comparing with complementary algorithms).**

Keywords-elliptic curve crytographic;public key; fast calculations in ECC, complementary recoding, 1's complement of binary numbers.

I. INTRODUCTION

Wireless networks, in particular sensor networks, have been experiencing an explosive growth in recent years and offered attractive flexibility to network operators and users.

Security in communication networks has become increasingly prominent and its key technology cryptography technology develops rapidly.

Elliptic curve cryptography (ECC) has been known for public-key cryptography purposes [1], [2], which is independently introduced by Koblitz and Miller in 80's has attracted increasing attention in recent years due to its linear scalability, a small software footprint, low hardware implementation cost, low bandwidth requirement, and high device performance. ECC that relies on the difficulty of elliptic curve discrete logarithmic problem (ECDLP) based cryptosystem is an efficient public key cryptosystem, which is more suitable for limited environments.

Normally the structure of an ECC operation involves three computational levels, namely scalar multiplication algorithm, point arithmetic and field arithmetic [3] mainly focus on improvements at the point arithmetic level to decreasing the time of ECC scalar multiplication. For point adding, a combination of projective and affine coordinates, i.e. mixed addition [4], has offered the efficient formulae. In the case of adding points in the same coordinate system, the required formula is more costly and is referred to as general addition. Recently, some approaches to compute faster scalar multiplications, such as double-base chain [5], ternary/binary method [6] have introduced tripling as a new point operation. Also there are many papers discussed the implementation over various situations [7-14] to try to get a better results. It is well known that ECC with a key size of 160 bits provides the same level of security as RSA, DSA, and DH with a key size of 1024 bits. The most important issue is that the standard bodies such as IEEE [16], [20], ISO, IETF etc. have accepted ECC as an alternative, efficient public key cryptosystem. It is the fact that the benefits of elliptic curve cryptography becomes an important member in the public key cryptosystem family with it's a small software footprint, low hardware implementation costs, linear scalability, low bandwidth requirements and high performance.

The most expensive operation in elliptic curve based cryptographic protocol is the scalar multiplication. There are many papers investigated this issue, such as ECC using modified complementary [15], binary method [16], non-adjacent form (NAF) [11], and mutual opposite form (MOF) [18] and complements method [19], etc.

In fact for the various algorithms the target is to reduce the expensive operation-scalar multiplication, which can be expressed by decreasing so-called Hamming weight of binary representation if the base is used in binary base that the sensor networks are widely used, which we are focusing on in this paper.

In this paper a novel algorithm is presented with hybrid efficient complementary and 1's complement of binary numbers to significantly minimize Hamming weight, which makes the calculations significantly decreasing. The final results with this proposed algorithm compared with the listed methods on [15] the average calculating time estimated to be decreased about 12.5%. Even there is a room to be further improved for the proposed algorithm in the next step for our future research project.

The next section will discuss the general tradition elliptic curve cryptography algorithm and then some related algorithms that were focusing on decreasing the calculations over ECC will be followed. After that, we are going to show the advantages and drawbacks for the complementary

algorithm and 1's complement of binary numbers algorithm, which leads the next section, the novel algorithm that significantly decreases the Hamming weight. The final section, conclusion, presents the comparison results to show the proposed algorithm at current level about average 12.5% calculation time is saved.

II. TRADITIONAL PROTOCOL IN ECC

An elliptic curve is the set of solutions of an equation of the form can be shown as below:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \quad (1)$$

where a, b, c, d, and e, are real numbers.

A special addition operation is defined over elliptic curves and this with the inclusion of a point $\mathcal{O}$, called point at infinity. If three points are on a line intersecting an elliptic curve, then their sum is equal to this point at infinity $\mathcal{O}$, which acts as the identity element for this addition operation. Sometimes the general equation (1) can be referred as Weierstrass equation as shown in (2):

$$y^2 = x^3 + ax + b \quad (2)$$

If we wanted use a elliptic curve to be used for cryptography the necessary condition is the curve is not singular, i.e. the discriminant of polynomial $f(x) = x^3 + ax + b$:

$$4a^3 + 27b^2 \neq 0 \quad (3)$$

Figures 1 and 2 show the two elliptic curves are

$$y^2 = x^3 + 2x + 5 \quad (4)$$

and

$$y^2 = x^3 - 2x + 1 \quad (5)$$

It is noted that they are in different "shapes', Figure 1 is only part as the whole elliptic curve but Figure 2 has two part as the whole elliptic curve. However, we can check those two equations meet equation (3).

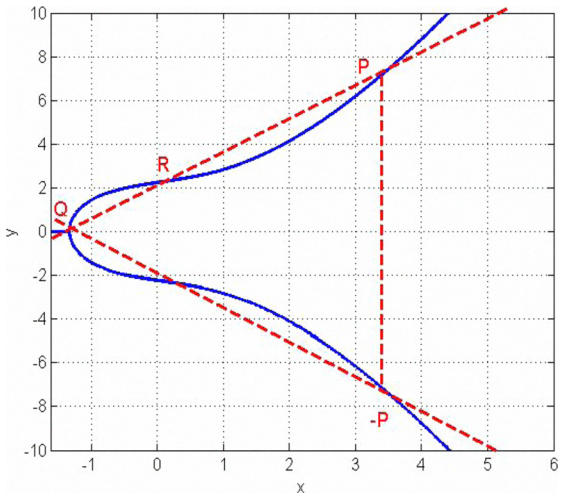

Figure 1. Elliptic curves equation (4)

An elliptic group over the Galois Field $E_p(a,b)$ is obtained by computing $x^3 + ax + b \bmod p$ for $0 \le x < p$. The constants a and b are non negative integers smaller than the prime number p and as here we used "mod p", so equation (3) should be read as:

$$4a^3 + 27b^2 \bmod p \neq 0 \quad (6)$$

For each value of x one needs to determine whether or not it is a quadratic residue. If it is the case, then there are two values in the elliptic group. If not, then the point is not in the elliptic $E_p(a,b)$ group.

When we fixed a prime number, p and then fixed constants a and b we can have the Galois Field $E_p(a,b)$ group.

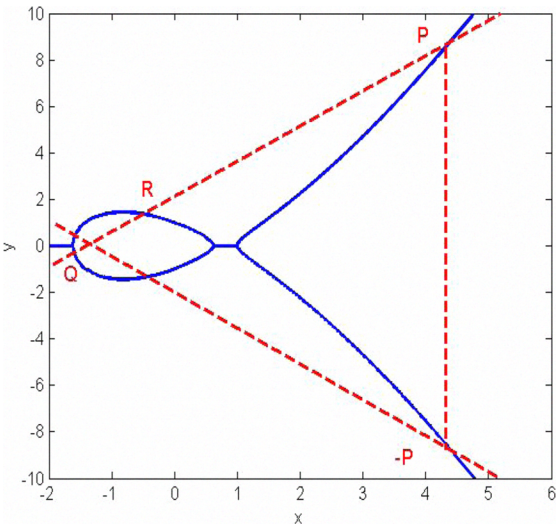

Figure 2. Elliptic curve eqaution (5)

For example, let the points $P = (x_1, y_1)$ and $Q\ (x_2, y_2)$ be in the elliptic group $E_p(a,b)$ group and $\mathcal{O}$ be the point at infinity. The rules for addition over the elliptic group $E_p(a,b)$ are :

(1) $P + \mathcal{O} = \mathcal{O} + P = P$

(2) If $x_2 = x_1$ and $y_2 = -y_1$, that is $P(x_1, y_1)$ and $Q = (x_2, y_2) = (x_1, -y_1) = -P$, that is the case: $P + Q = \mathcal{O}$.

(3) If $Q \neq -P$, then their sum $P + Q = (x_3, y_3)$ is given by ;

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p \quad (7)$$

where $$\lambda \equiv \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } P = Q \end{cases} \quad (8)$$

In order to express our new protocol of the hidden generator point we, without losing generality, use an example for the above description. Let's assume $p = 23$ and $a = 1$ and $b = 1$, i.e. the equation becomes: $y^2 = x^3 + x + 1 \bmod 23$. We have $4a^3 + 27b^2 \bmod 23 = 8 \neq 0$. Now we need to determine if y^2 is in the set of quadratic residues or not.

Now let's have a closer look at the elliptic group $E_p(a,b)$. In our above example, we pick the prime number $p = 23$ (it is noted that this is only for explaining the new protocol, in real life the p is bigger than this), we have quadratic residues

group $(p\text{-}1)/2 = 11$ and for this group the $E_p(a,b)$ can be shown as below:

$$E_{23}(1,1)=\begin{Bmatrix} (0,1) & (0,22) & (1,7) & (1,16) & (3,10) & (3,13) & (4,0) \\ (5,4) & (5,19) & (6,4) & (6,19) & (7,11) & (7,12) & (9,7) \\ (9,16) & (11,3) & (11,20) & (12,4) & (12,19) & (13,7) & (13,16) \\ (17,3) & (17,20) & (18,3) & (18,20) & (19,5) & (19,18) & \end{Bmatrix} \quad (9)$$

which includes the point (4, 0) corresponding to the single value $y = 0$. As an example, paper [22] discussed this in details regarding to the hidden generator point for protecting from man-in-the middle.

The elliptic curve cryptography can be used to encrypt plaintext messages, M, into ciphertexts. The plaintext message M is encoded into a point P_M from the finite set of points in the elliptic group, $E_p(a,b)$. First step consists in choosing a generator point, $G \in E_p(a,b)$, such that the smallest value of n for which $nG = \mathcal{O}$ is a very large prime number. Normally the traditional ECC protocol is let the elliptic group $E_p(a,b)$ and the generator point G be in public. Each user select a private key, say $n_A < n$ and compute the public key P_A as $P_A = n_A G$. Then, the case becomes encrypting the message point P_M for the partner, say from Alice to Bob. So Alice (A) chose a random integer k and computes the ciphertext pair of points P_C using Bob's public key P_B:

$$P_C = [(kG),(P_M + kP_B)] \quad (9)$$

Bob received the ciphertext pair of points, P_C then multiplies the first point, (kG) with his private key, n_B, and then adds the result to the second point in the ciphertext pair of points as shown below:

$$(P_M + kP_B) - [n_B(kG)] = P_M \quad (10)$$

which is the plaintext point, corresponding to the plaintext message M. It is noted that only Bob can obtain retrieve the plantext information P_M by the private key n_B. The cryptographic strength of ECC lies in the difficulty for a cryptanalyst to determine the secret random number k from kP and P itself. The fast method to solve this problem is known as the elliptic curve logarithm problem (ECLP).

III. Speeding up Algorithms over ECC

The scalar multiplication is very expensive operation in elliptic curve based cryptographic protocol. Hence, the speed of scalar multiplication plays an important role in the efficient system.

Scalar multiplication is the computation of the form $Q = kP$, where P and Q are the elliptic curve points (as figures shown) and k is an integer. It can be obtained by repeated elliptic curve point addition and doubling operations. In the binary algorithms [16], the integer k is represented as

$$k = \sum_{j=0}^{l-1} k_j 2^j \text{ where } k_j \in \{0,1\}, \quad (11)$$

which scans the bits of k either from left-to-right or right-to-left. The cost of multiplication depends on the length of the binary representation of k and the number of Harming weight of scalar representation in this representation. If the representation $(k_{n-1}\ldots k_0)_2$ with $k_{n-1} \neq 0$ then the number of doubling operation is $(n\text{-}1)$. In an average, binary algorithm requires $(n\text{-}1)$ doublings and $(n\text{-}1)/2$ additions. For example, $k = 1778$, then $k = (11011111100)_2$ so computation of 1778 P requires 10 doublings and 5 additions.

It is well known that a algorithm called non-adjacent form (NAF) [17], based on the fact that k is represented as $k = \sum_{j=0}^{l-1} k_j 2^j$, with $k_j \in \{-1,0,1\}$. which using three digits $\{0, 1, -1\}$-radix 2 representation and this conversion is taken from right-to-left. The average Hamming weight of signed binary representation is $n/3$ and it has the lower Hamming weight than the binary algorithm. However, it is noted the Hamming weight is one of keys to handle computation load, for example, $k = 255$, or $(11111111)_2$, computation of $255P$ requires 7 point additions, but if it is transformed by $(10000000\text{-}1)P$, which is $256P\text{-}P$, only one addition is required.

There is another algorithm needs to be mentioned, the mutual opposite form (MOF) [18], which converts the binary string to MOF from the most significant bit efficiently. The n-bit binary string k is converted into a signed binary string by $mk = 2k - k$, with "–" stands for a bit subtraction. The conversion of MOF representation of an integer is highly flexible because conversion can be made either from right-to-left or left-to-right. The output of MOF is comparable efficiency with out of NAF as shown in [15].

IV. Complementary and the 1's Complement of Binary Numbers

As above described it is clearly to see that every mentioned algorithm makes the target that decreasing the Hamming weight to increase the efficient computation over ECC. As [15] shown that the MOF and complementary algorithms are almost the same level in terms of computation costs we may take complementary algorithm as part of hybrid algorithm as shown below. But we need to present the so-called "the 1's complement of binary numbers" described by Gillie [21].

The 1's complement of any binary number may be found by the following equation [21]:

$$C_1 = (2^a - 1) - k \quad (12)$$

Or $$k = (2^a - 1) - C_1 \quad (13)$$

where C_1 = 1's complement of the number
a = number of digits to be handled by the computer
k = binary number whose 1's complement

As an example, let $k = 1788$, or $k = (11011111100)_2$ in its binary form. C_1 = 1's Complement of the number of k and the a in this example it is in binary form is 11. Therefore from the equation (12) we have:

$$C_1 = (2^a - 1) - k = (2^{11} - 1) - (11011111100)$$
$$= 00100000011$$

Therefore we

$$k = 1788 = (2^a - 1) - C_1$$
$$= (2^{11} - 1) - 00100000011$$

This means $k = 1788 = (100000000000 - 00100000011 - 1)_2$, which gives :

$$1788 = 2048 - 256 - 2 - 1 - 1$$

Hence, this shows that the Hamming Weight of scalar k has reduced from original 8 to current 5 which will save 3 elliptic curve addition operations. One addition operation requires 2 Squaring, 2 Multiplication and 1 inverse operation. But if the original binary form of k is critical for this method as if the number of 1s in original binary form of k is > the one-half of the bit's length, i.e. 1's number $\geq a/2$ then there is no need to convert the original binary format into "1's complement format" as our target is to decrease Harming weight. So our proposed algorithm is that (1) check the 1's number of the binary form, if it is $\geq a/2$, then go to the "complementary algorithm" [15], if it is not, then go to "1's complement format", i.e. go to the equation (12) then go to equation (13).

It is noted that here there is a checking processing before go head for which way to calculate the scale multiplication, which there is time costs but as the either way for the computation of the scale multiplication is the most efficient due to the minimizing Harming weight the saved time can pay the checking costs. In fact the final results, shown in the next section by the table, support this conclusion due to the checking processing is almost costing nothing in comparison with the saved time when the Harming weight is minimized.

V. CONCLUSION

The importance of security in communication system has become increasingly prominent, and its major technology cryptography technology develops rapidly. ECC has become an important branch of public key cryptography system as it has many benefits for the devices for wireless network, which has restrictions of the limited bandwidth, processing power, and storage space and power consumption.

The efficiency of ECC implementation is highly dependent on the performance of arithmetic operations of scalar multiplication. This paper based on discussions of the current major algorithms present a novel algorithm, hybrid of the "1's complement of binary number" and "complementary" to minimizing Harming weight to speed up the calculation over ECC. The final results are summarized in the table shown in below, where the results obtained from [15] was used.

In terms of average, the proposed algorithm is about 12.5% saved time in comparison with the results of complementary algorithm in [15].

As we have seen that due to the checking processing, there is always the case that the Hamming Weight will less than the half of the length (in terms of digit number) and the either complement of the number method or 1's complement of the number method will constantly keep the Hamming Weight minimizing, which makes this method sitting on the very power saving position due to the computing works.

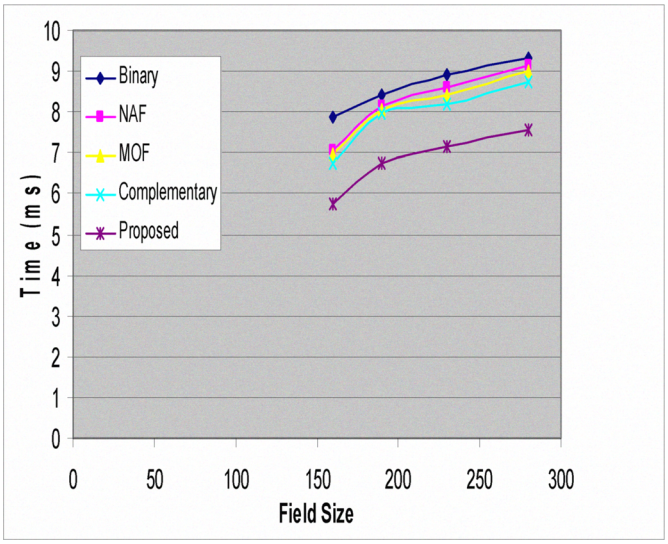

Figure 3. Comparison the proposed algorithm with the nominated algorithms by [15] (the data for the nominated algorithms were used from the same reference).

REFERENCES

[1] V.S. Miller, "uses of elliptic curves in cryptography," in Advances in Cryptology, CRYPTO'85, ser . Lecture Notes in Computer Science, vol. 218, Springer, 1986. pp. 417-428.

[2] N. Koblitz, " Elliptic curve cryptosystems," Mathematics of Compution, vol. 48, no.177, pp.203-209, Jan 1987.

[3] D. Hakerson, A. Menezes, and S. Vanston, "Guide to Elliptic Curve Cryptography," Springer-Verlag, NY (2004).

[4] H. Cohen, A Miyaji and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Lectures Notes in Computer Science, 1514, 51-65 (1998).

[5] V. Dimitrov V., L. Imbert, and P. K. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," Lectures Notes in Computer Science, 3788, 59-78 (2005).

[6] M. Ciet, M. Joye, K. Lauter and P.L. Montgomery,"Trading inversions for multiplications in elliptic curve cryptography," Designs, Codes, and Cryptography, 39, 189-206 (2006).

[7] D. Bernstein, "High-speed diffie-hellman, part 2," presented at the INDOCRYPT'06 tutorial session, Dec. 11-13, Kolkata, India (2006).

[8] K. Kaabneh and H. Al-Bdour, "Key exchange protocol in elliptic curve cryptography with no public point," American Journal of Applied Sciences 2 (8): 1232-1235, 2005.

[9] J. Adikari, V. Dimitrov, and L. Imbert, "Hybrid binary-ternary joint sparse from and its application in ellipic curve cryptography," Cryptology ePrint Archive, Report 2008/285, 2008.

[10] Bangju Wang, Huanguo Zhang and Yuhua Wang, "An efficient elliptic curves scalar multiplication for wireless network," 2007 IFIP International Conference on Network and Parallel Computing-Workship, pp131.

[11] Shiwei Ma, Yuanling Hao, Zhongqiao Pan, and Hui Chen, "Fast implementation for modular inversion and scalar multiplication in the

elliptic curve cryptography," 2008 Second International Symposium on Intelligent Information Technology Application, pp488.

[12] Michael Scott, "Optimal Irreducible Polynomials for GF(2^m) Arithmetic," Cryptology ePrint Archive, Report 2007/192, 2007.

[13] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks," International Journal of Security and Networks, vol. 1, no.3/4, 2006.

[14] A. Liu, P. Kampanakis, and P. Ning, "TinyECC: Elliptic curve cryptography for sensor networks ," (version 10), november 2007.

[15] P. Balasubramaniam and E. Karthikeyan, "Elliptic curve scaslar multiplication algorithm using complementary recoding," Applied Mathematics and Computer, 2007 pp.1-6. doi: 10.1016/j.amc.2007.01.015.

[16] Standard Specifications for Public Key Cryptography, IEEE standard 1363, 2000.

[17] F. Morain and J. Olivos, "Speeding up the computations on an elliptic curve using addition-subtraction chains," RAIRO Theoretical Informatics and Applications 24 (1990) pp.531 –pp.543.

[18] K. Okeya, "Signed binary representations revisited," Proceedings of CRYPTO'04 (2004) pp123-139.

[19] C. C. Chang, Y.T. Kuo, and C. H. Lin, "Fast algorithms for common multiplicand multiplication and exponentiation by performing complements," Proceeding of 17th International Conference on Advanced Information Networking and Applications, March, 2003, pp.807-811.

[20] Kossi Edoh, "Elliptic Curve Cryptography on PocketPCs," International Journal of Security and Its Applications, Vol.3 No.3, July 2009, pp.23-33

[21] Angelo C Gillie, "Binary Arithmetic and Boolean algebra," McGRAW-HILL Book Company, 1965. pp53.

[22] Xu Huang, Pritam Gajkumar Shah, and Dharmendra Sharma, "Elliptic Curve Crytography for Public Key with Hidden Generator Point," IEEE, ElectroRem09, 30 November-02 December 2009, Melbourne, Australia.